Boosting the Margin

An Explanation for the Effectiveness of Voting Methods?

Lev Reyzin

Clique Talk, Spring '07

The Papers*

- Schapire, R. E. (2002). **The boosting approach to machine learning: An overview.** Nonlinear Estimation and Classification. Springer. (Covers much of his work with Yoav Freund)
- Schapire, R. E., Freund, Y., Bartlett, P., & Lee, W. S. (1998). **Boosting the margin: A new explanation for the effectiveness of voting methods.** The Annals of Statistics, 26, 1651–1686.
- Breiman, L. (1998). **Arcing classifiers.** The Annals of Statistics, 26, 801–849.
- Lev Reyzin and Robert E. Schapire. **How Boosting the Margin Can Also Boost Classifier Complexity.** In Proceedings of the 23rd Conference on Machine Learning (ICML), June 2006

* Some material on these slides is taken directly from the papers above and from http://www.cs.princeton.edu/courses/archive/spring03/cs511/

The Learning Task

- Given training examples and their labels
- Predict the label of new test examples chosen from the same distribution as the training data

Some Definitions

Training Data: labeled examples given to a learner

Test Data: examples whose label a learner must predict

Training Error: the prediction error of the final hypothesis on the training data

Generalization Error: the true prediction error of the final hypothesis on new data.

Test Error: the prediction error of the final hypothesis on the test data (an estimate of the generalization error)

Hypothesis: the prediction rule a learner forms based on training data to predict on new data

An Example of the Task

Training data:

(1,1,0,0,1) -> 1 (0,0,0,0,1) -> 0

(1,0,0,1,1) -> 0 (0,0,1,0,0) -> 0

(0,1,0,0,1) -> 1 (0,1,1,1,1) -> 1

…

Test data:

(1,1,1,1,1) (0,1,1,1,0)

(1,0,0,0,0) (0,0,1,1,1)

Overfitting

Training Data

(1,1,0,0,1) -> 1	(0,0,0,0,1) -> 0
(1,0,0,1,1) -> 0	(0,0,1,0,0) -> 0
(0,1,0,0,1) -> 1	(0,1,1,1,1) -> 1

Rule 1:

(x1x2-x3-x4x5) v (-x1x2-x3-x4x5) v (-x1x2x3x4x5)

Rule 2:

(x2)

Occam's Razor says we should pick rule 2
Rule 2 comes from a smaller hypothesis space
Rule 1 overfits the training data

The Idea of Boosting

- Combine many “moderately inaccurate” base classifiers (do better than chance) into a combined predictor (that predicts arbitrarily well)
- Generate a new base classifier in each round
- Constantly focus on the hardest examples
- The final predictor is the weighted vote of the base classifiers

The Main Characters

- x = a training example
- y = its label
- T = the number of rounds of boosting
- t = the current round of boosting
- m = the number of training examples
- D = the weight distribution on training examples
- h = the hypothesis
- ϵ = the error of the hypothesis
- α = the voting weight of the hypothesis
- d = the vc dimension of the base learner

More Formally…

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

AdaBoost (Freund, Schapire)

$$\boxed{\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \epsilon_t}{\epsilon_t} \right)}$$

- Train base learner using distribution D_t.
- Get base classifier $h_t : X \to \mathbb{R}$.
- Choose $\alpha_t \in \mathbb{R}$.
- Update:

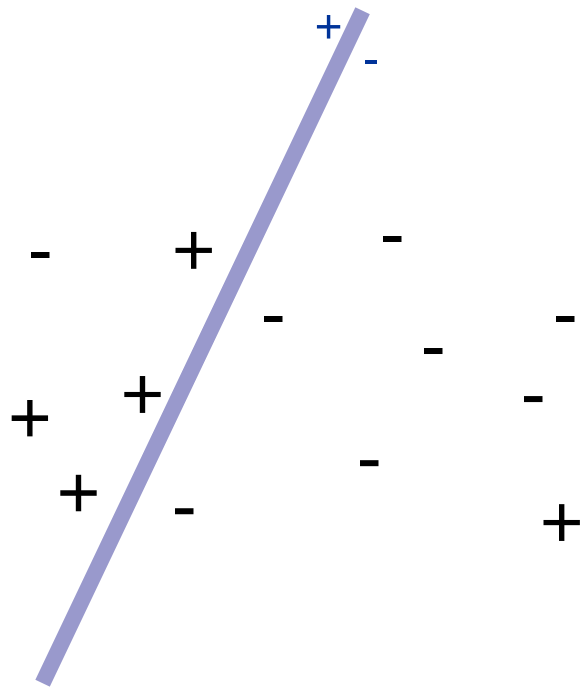
$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

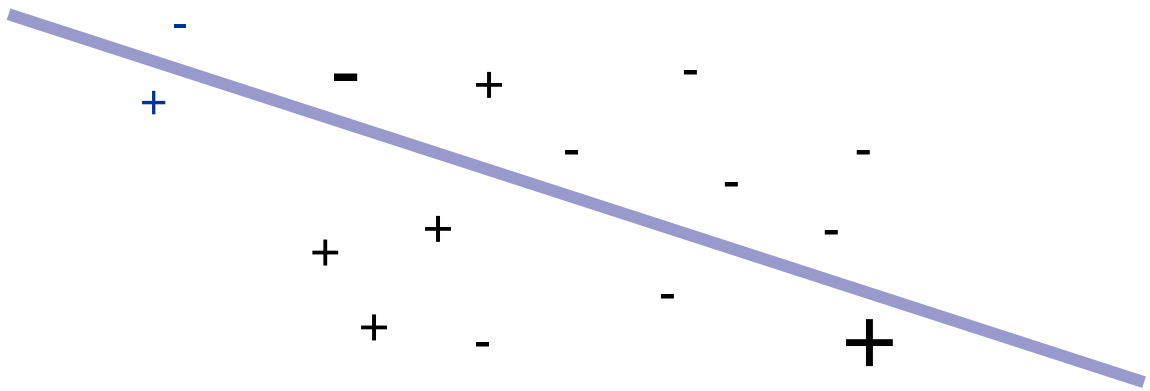
Output the final classifier:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x) \right).$$

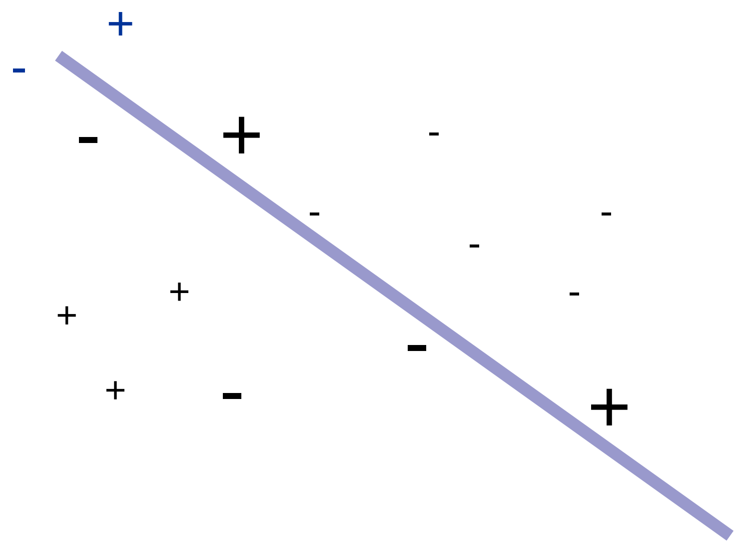
An Example



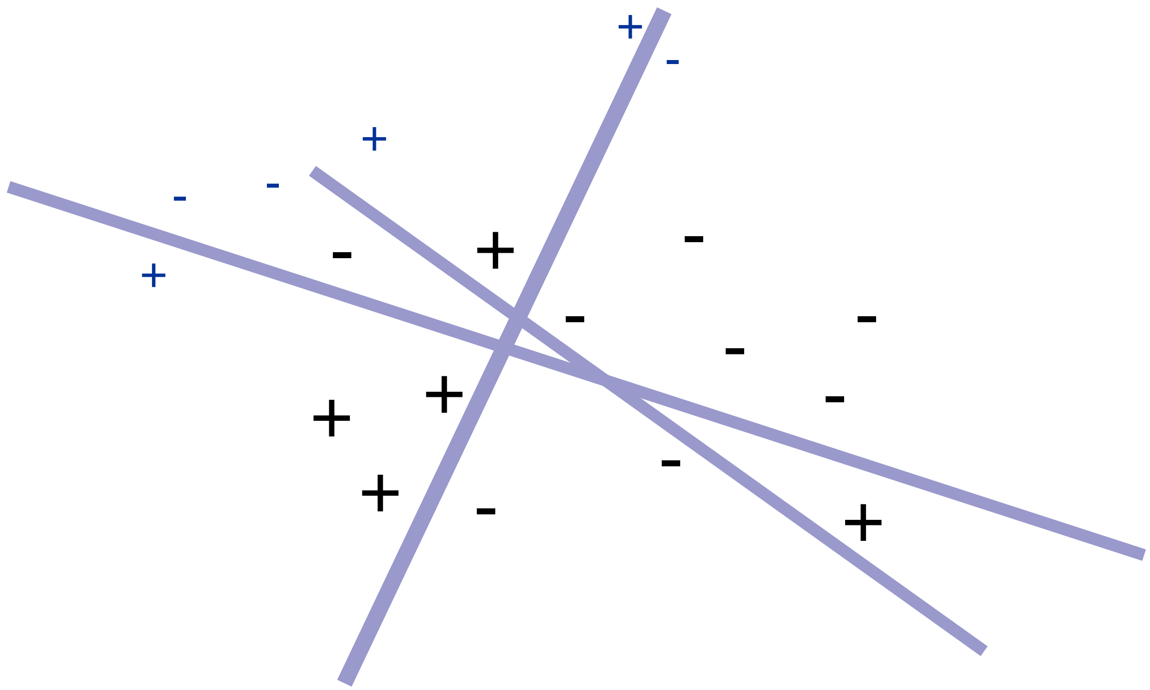
An Example



An Example



An Example



We classified our training data correctly!

But wait – what did we accomplish?

Relating to Generalization Error

(Freund and Schapire)
whp, the generalization error is less than:

$$\hat{\Pr}\left[H(x) \neq y\right] + \tilde{O}\left(\sqrt{\frac{Td}{m}}\right)$$

empirical probability of getting a training example wrong

hiding log factors

Bounding the Empirical Training Error

Theorem: $\hat{\Pr}[H(x) \neq y] \leq \prod_t Z_t$

Lemma: $D_{T+1}(x_i) = \dfrac{\exp(-y_i f(x_i))}{m \prod_t Z_t}$ where $f(x_i) = \sum_{t=1}^{T} \alpha_t \cdot h_t(x_i)$

$$
\begin{aligned}
D_{T+1}(i) &= \frac{D_T(i) \cdot \exp(-\alpha_T y_i h_T(x_i))}{Z_T} \\
&= \frac{D_{T-1}(i) \cdot \exp(-\alpha_{T-1} y_i h_{T-1}(x_i)) \cdot \exp(-\alpha_T y_i h_T(x_i))}{Z_{T-1} \cdot Z_T} \\
&\vdots \\
&= \frac{1}{m} \cdot \frac{\exp(-y_i \cdot \sum_t \alpha_t h_t(x_i))}{\prod_t Z_t}
\end{aligned}
$$

Bounding Training Error (continued)

$$
\begin{aligned}
\hat{\Pr}\left[H(x) \neq y\right] &= \frac{1}{m} \cdot \sum_{i=1}^{m} [\![y_i \neq H(x_i)]\!] \\
&= \frac{1}{m} \cdot \sum_{i=1}^{m} [\![y_i f(x_i) \leq 0]\!] \\
&\leq \frac{1}{m} \sum_{i=1}^{m} e^{-y_i f(x_i)} \\
&= \frac{1}{m} \cdot \sum_{i=1}^{m} D_{T+1}(i) \cdot m \cdot \prod_t Z_t \\
&= \frac{1}{m} \prod_t Z_t \cdot \sum_{i=1}^{m} D_{T+1}(i) \cdot m \\
&= \prod_t Z_t.
\end{aligned}
$$

$\square$

Choosing Alpha

$$\begin{aligned} Z_t &= \sum_{i=1}^{m} D_t(i) \cdot e^{-\alpha_t y_i h_t(x_i)} \\ &= \sum_{i:h_t(x_i)\neq y_i} D_t(i) \cdot e^{\alpha_t} + \sum_{i:h_t(x_i)=y_i} D_t(i) \cdot e^{-\alpha_t} \\ &= \epsilon_t \cdot e^{\alpha_t} + (1-\epsilon_t) \cdot e^{-\alpha_t}. \end{aligned}$$

so if we choose alpha so that Z_t is minimized, we get AdaBoost

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

Training Error Drops Exponentially

we define gamma to be the "edge," or how much better than random a weak learner is performing:

$$\gamma_t = 1/2 - \epsilon_t.$$

then our choice of alpha gives:

$$\prod_t Z_t = \prod_t \left[2\sqrt{\epsilon_t(1-\epsilon_t)}\right] = \prod_t \sqrt{1-4\gamma_t^2} \le \exp\left(-2\sum_t \gamma_t^2\right)$$

therefore the training error falls exponentially in T:

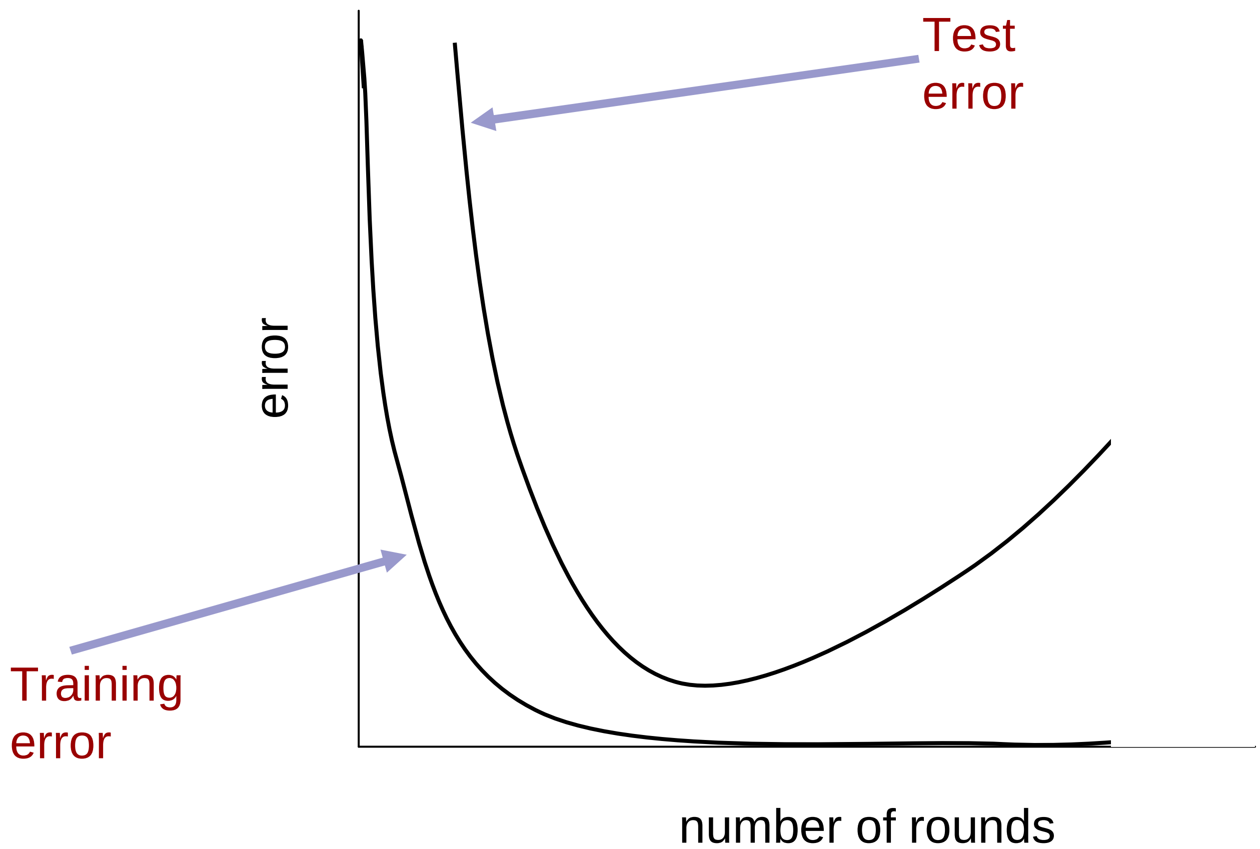
$$\hat{\Pr}\left[H(x) \neq y\right] \le \prod_t Z_t \le e^{-2T\gamma^2}$$

Back to the Bound

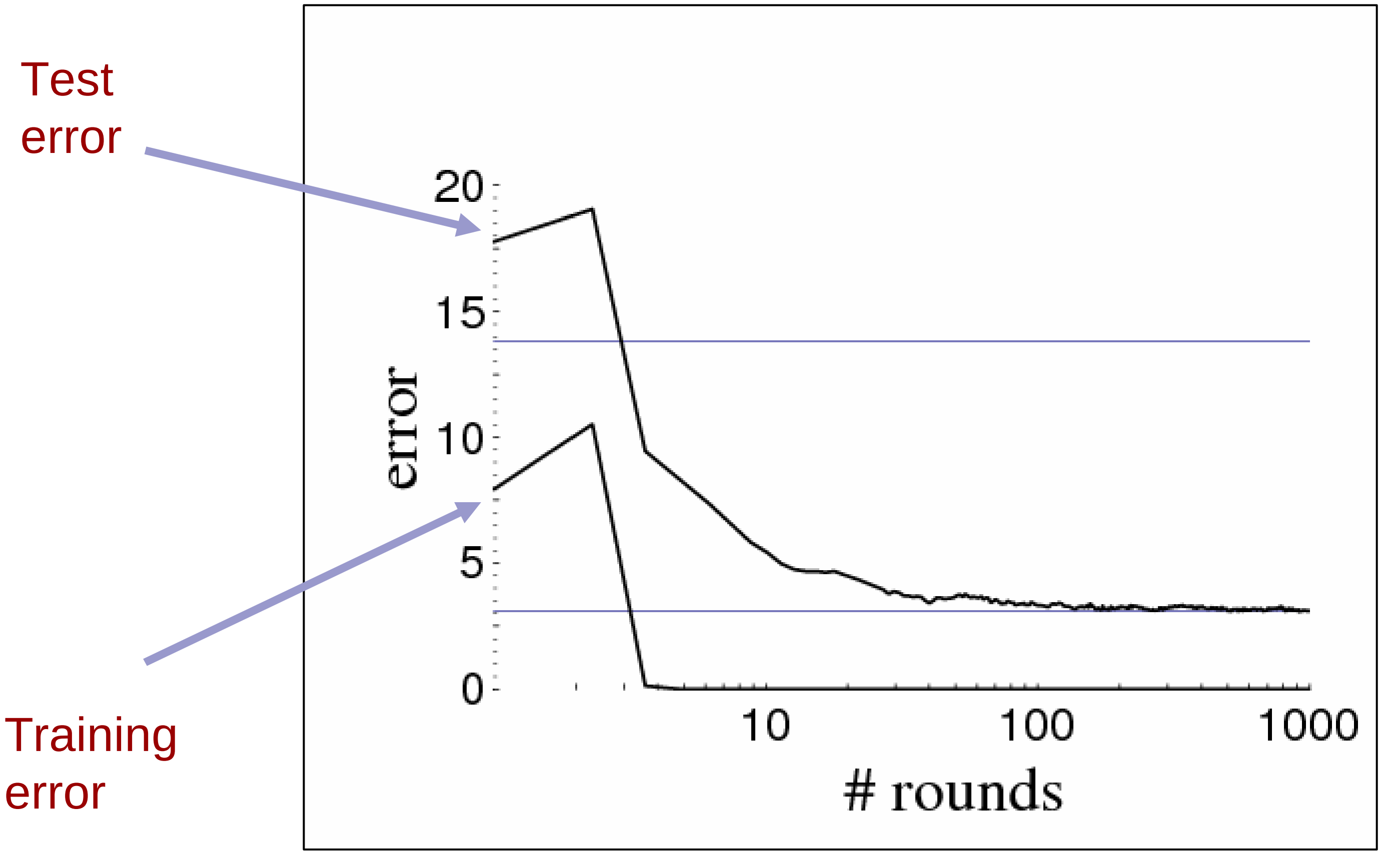
whp, the generalization error is less than:

$$\hat{\Pr}\left[H(x) \neq y\right] + \tilde{O}\left(\sqrt{\frac{Td}{m}}\right)$$

We Would Expect Overfitting



However…



[Drucker & Cortes; Breiman; Quinlan, …]

The Margin

- The margin of a classifier on an example:
  - margin = (weighted fraction of base classifiers voting for correct label) – (weighted fraction voting for incorrect label)
  - magnitude represents the confidence of the vote
  - positive if the vote gives the correct classification. Otherwise it's negative.
  - margin on example i = $y_i f(x_i)$ where $f(x_i) = \sum_{t=1}^{T} \alpha_t \cdot h_t(x_i)$
- Margins are measured over <u>training</u> examples

A Margin Bound

- A later bound relied on the margins the classifier achieved on the training examples and not on the number of rounds of boosting. [Schapire et. al. '98]

the generalization error is at most:

the VC dimension of the base classifier

$$\underbrace{\hat{\Pr}\left[\text{margin}_f(x,y) \leq \theta\right]}_{} + \tilde{O}\left(\sqrt{\frac{d}{m\theta^2}}\right)$$

Fraction training examples with margin below theta

number of training examples

for any value of theta

Proof *(sketch)* of Margin Bound

We define the convex hull C to be the set of mappings that can be generated by taking a weighted average if classifiers from *H*

$$\mathcal{C} \doteq \left\{ f : x \mapsto \sum_{h \in \mathcal{H}} a_h h(x) \,\middle|\, a_h \geq 0;\ \sum_h a_h = 1 \right\}$$

We define C_N to be the set of unweighted averages over N elements from *H*

$$\mathcal{C}_N \doteq \left\{ f : x \mapsto \frac{1}{N} \sum_{i=1}^{N} h_i(x) \,\middle|\, h_i \in \mathcal{H} \right\}$$

We use $P_{(x,y)-D}[A]$ to denote the probability of the event A when the example (x,y) is chosen according to D (the distribution from which examples are generated). This is abbreviated $P_D[A]$

We use $P_{(x,y)-S}[A]$ to denote the probability with respect to choosing an example uniformly at random from the training set. This is abbreviated $P_S[A]$

Proof of Margin Bound (part 2)

We let f be a majority vote classifier from C.

By choosing N elements independently at random according to this distribution, we can generate an element of C_N.

A function g in C_N distributed according to Q is selected by choosing $h_1,\ldots,h_N$ at random according to coefficients a_h.

Since for any from events A and B

$$\mathbf{P}[A] = \mathbf{P}[B \cap A] + \mathbf{P}\left[\overline{B} \cap A\right] \leq \mathbf{P}[B] + \mathbf{P}\left[\overline{B} \cap A\right]$$

We have

$$\mathbf{P}_{\mathcal{D}}[yf(x) \leq 0] \leq \mathbf{P}_{\mathcal{D}}[yg(x) \leq \theta/2] + \mathbf{P}_{\mathcal{D}}[yg(x) > \theta/2, yf(x) \leq 0]$$

since this holds for any g, we can take exp val of rhs wrt Q and get

$$\begin{aligned}
&\leq \mathbf{P}_{\mathcal{D},g\sim Q}[yg(x) \leq \theta/2] + \mathbf{P}_{\mathcal{D},g\sim Q}[yg(x) > \theta/2, yf(x) \leq 0] \\
&= \mathbf{E}_{g\sim Q}[\mathbf{P}_{\mathcal{D}}[yg(x) \leq \theta/2]] + \mathbf{E}_{\mathcal{D}}[\mathbf{P}_{g\sim Q}[yg(x) > \theta/2, yf(x) \leq 0]] \\
&\leq \mathbf{E}_{g\sim Q}[\mathbf{P}_{\mathcal{D}}[yg(x) \leq \theta/2]] + \mathbf{E}_{\mathcal{D}}[\mathbf{P}_{g\sim Q}[yg(x) > \theta/2 \mid yf(x) \leq 0]]
\end{aligned}$$

Proof of Margin Bound (part 3)

from the previous slide, we have

$$\mathbf{P}_{\mathcal{D}}\left[yf(x) \leq 0\right] \leq \quad \mathbf{E}_{g\sim\mathcal{Q}}\left[\mathbf{P}_{\mathcal{D}}\left[yg(x) \leq \theta/2\right]\right] + \mathbf{E}_{\mathcal{D}}\left[\mathbf{P}_{g\sim\mathcal{Q}}\left[yg(x) > \theta/2 \mid yf(x) \leq 0\right]\right]$$

we now bound both terms on the rhs separately.

Since $f(x) = \mathbf{E}_{g\sim\mathcal{Q}}\left[g(x)\right]$, the probability in the expectation is that the avg over N draws is larger than its expected value by more than ϴ/2. A Chernoff bound yields:

$$\mathbf{P}_{g\sim\mathcal{Q}}\left[yg(x) > \theta/2 \mid yf(x) \leq 0\right] \leq e^{-N\theta^2/8}$$

For the first term we use the union bound (and a Chernoff bound). The probability over choices of S that there is a g and ϴ for which

$$\mathbf{P}_{\mathcal{D}}\left[yg(x) \leq \theta/2\right] > \mathbf{P}_{S}\left[yg(x) \leq \theta/2\right] + \epsilon_N$$

is at most

$$(N+1)|\mathcal{C}_N|e^{-2m\epsilon_N^2}$$

Chernoff bound

bound on the number of such choices

Proof of Margin Bound (part 4)

so if we set

$$\epsilon_N = \sqrt{(1/2m)\ln((N+1)|\mathcal{H}|^N/\delta_N)}$$

we take expectation wrt Q, we get that with probability $1-\delta_N$

$$\mathbf{P}_{\mathcal{D},g\sim\mathcal{Q}}\left[yg(x)\le\theta/2\right] \le \mathbf{P}_{S,g\sim\mathcal{Q}}\left[yg(x)\le\theta/2\right] + \epsilon_N$$

To finish the argument, we relate the fraction of the training set for which yg(x) ≤ ϴ/2 to the probability that yf(x) ≤ ϴ. We do this by the technique from the beginning.

$$\mathbf{P}[A] = \mathbf{P}[B\cap A] + \mathbf{P}\left[\overline{B}\cap A\right] \le \mathbf{P}[B] + \mathbf{P}\left[\overline{B}\cap A\right]$$

$$\begin{aligned}
\mathbf{P}_{S,g\sim\mathcal{Q}}\left[yg(x)\le\theta/2\right] &\le \mathbf{P}_{S,g\sim\mathcal{Q}}\left[yf(x)\le\theta\right] + \mathbf{P}_{S,g\sim\mathcal{Q}}\left[yg(x)\le\theta/2, yf(x)>\theta\right] \\
&= \mathbf{P}_S\left[yf(x)\le\theta\right] + \mathbf{E}_S\left[\mathbf{P}_{g\sim\mathcal{Q}}\left[yg(x)\le\theta/2, yf(x)>\theta\right]\right] \\
&\le \mathbf{P}_S\left[yf(x)\le\theta\right] + \mathbf{E}_S\left[\mathbf{P}_{g\sim\mathcal{Q}}\left[yg(x)\le\theta/2 \mid yf(x)>\theta\right]\right]
\end{aligned}$$

Proof of Margin Bound (part 5)

from the previous slide we have:

$$\mathbf{P}_{S,g\sim\mathcal{Q}}\left[yg(x) \le \theta/2\right] \le \quad \mathbf{P}_S\left[yf(x) \le \theta\right] + \mathbf{E}_S\left[\mathbf{P}_{g\sim\mathcal{Q}}\left[yg(x) \le \theta/2 \mid yf(x) > \theta\right]\right]$$

Again, using Chernoff bounds, we have:

$$\mathbf{P}_{g\sim\mathcal{Q}}\left[yg(x) \le \theta/2 \mid yf(x) > \theta\right] \le e^{-N\theta^2/8}$$

If we set $\delta_N = \delta/(N(N+1))$ and if we combine the equations above (and before), we get that for all $\Theta > 0$ and $N \ge 1$ with probability at least $1-\delta$

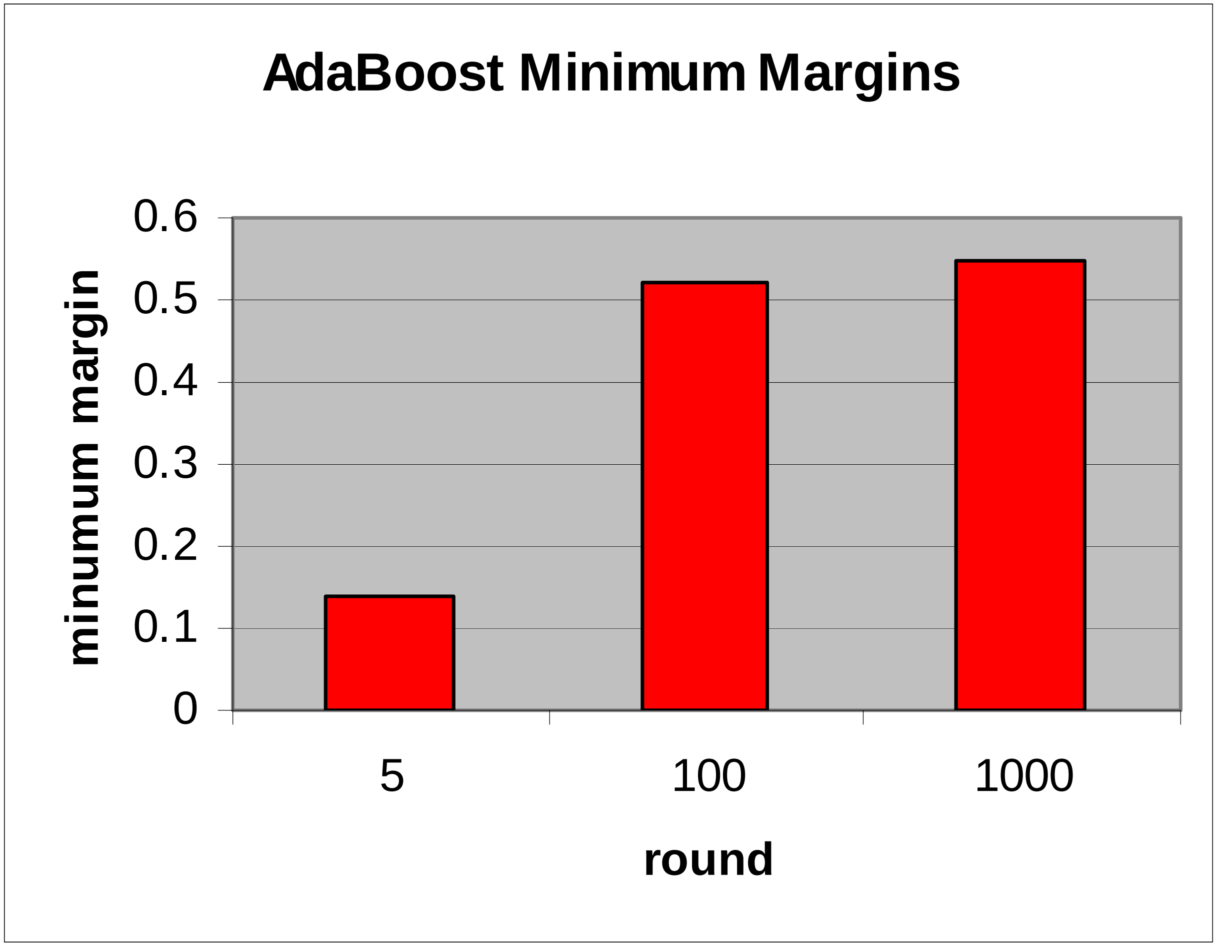
$$\mathbf{P}_{\mathcal{D}}\left[yf(x) \le 0\right] \le \mathbf{P}_S\left[yf(x) \le \theta\right] + 2e^{-N\theta^2/8} + \sqrt{\frac{1}{2m}\ln\left(\frac{N(N+1)^2|\mathcal{H}|^N}{\delta}\right)}$$

setting $N = \lfloor (4/\theta^2)\ln(m/\ln|\mathcal{H}|) \rfloor$ gives

$$\mathbf{P}_{\mathcal{D}}\left[yf(x) \le 0\right] \le \mathbf{P}_S\left[yf(x) \le \theta\right] + O\left(\frac{1}{\sqrt{m}}\left(\frac{\log m \log|\mathcal{H}|}{\theta^2} + \log(1/\delta)\right)^{1/2}\right)$$

$\square$

AdaBoost's Minimum Margins



The Margins Explanation

$$\underbrace{\hat{\Pr}\left[\mathrm{margin}_f(x,y) \leq \theta\right]}_{\text{Fraction training examples with margin below theta}} + \tilde{O}\left(\sqrt{\frac{d}{m\theta^2}}\right)$$

the VC dimension of the base classifier

number of training examples

for any value of theta

- AdaBoost pushes the cumulative margins distribution towards higher margins.

- All things being equal, higher margins mean lower generalization error.

arc-gv [Breiman '98]

- motivated by the margins explanation
  - arc-gv's minimum margin provably converges to the optimal
  - one line difference from AdaBoost

$$\alpha_t = \frac{1}{2}\log\frac{1-\epsilon_t}{\epsilon_t} - \frac{1}{2}\log\frac{1+\varrho}{1-\varrho}$$

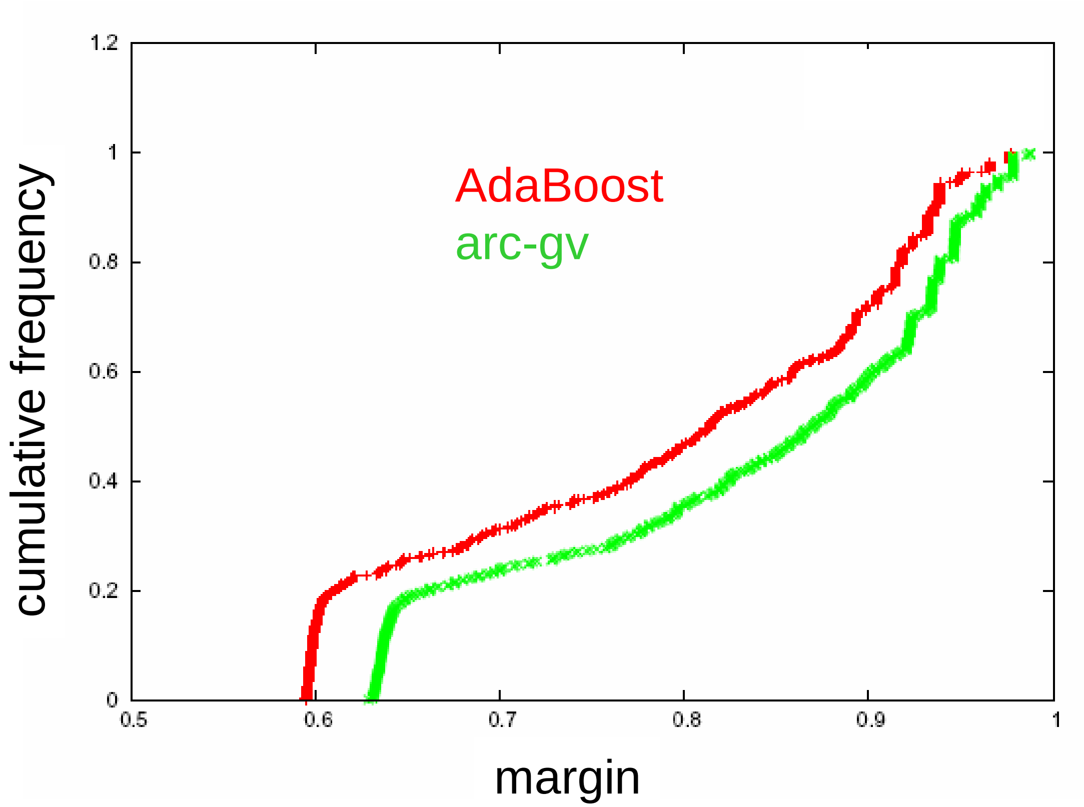
the minimum margin on any example of the combined vote thus far

- Breiman's reasoning: higher minimum margin would imply lower test error

The Experiments

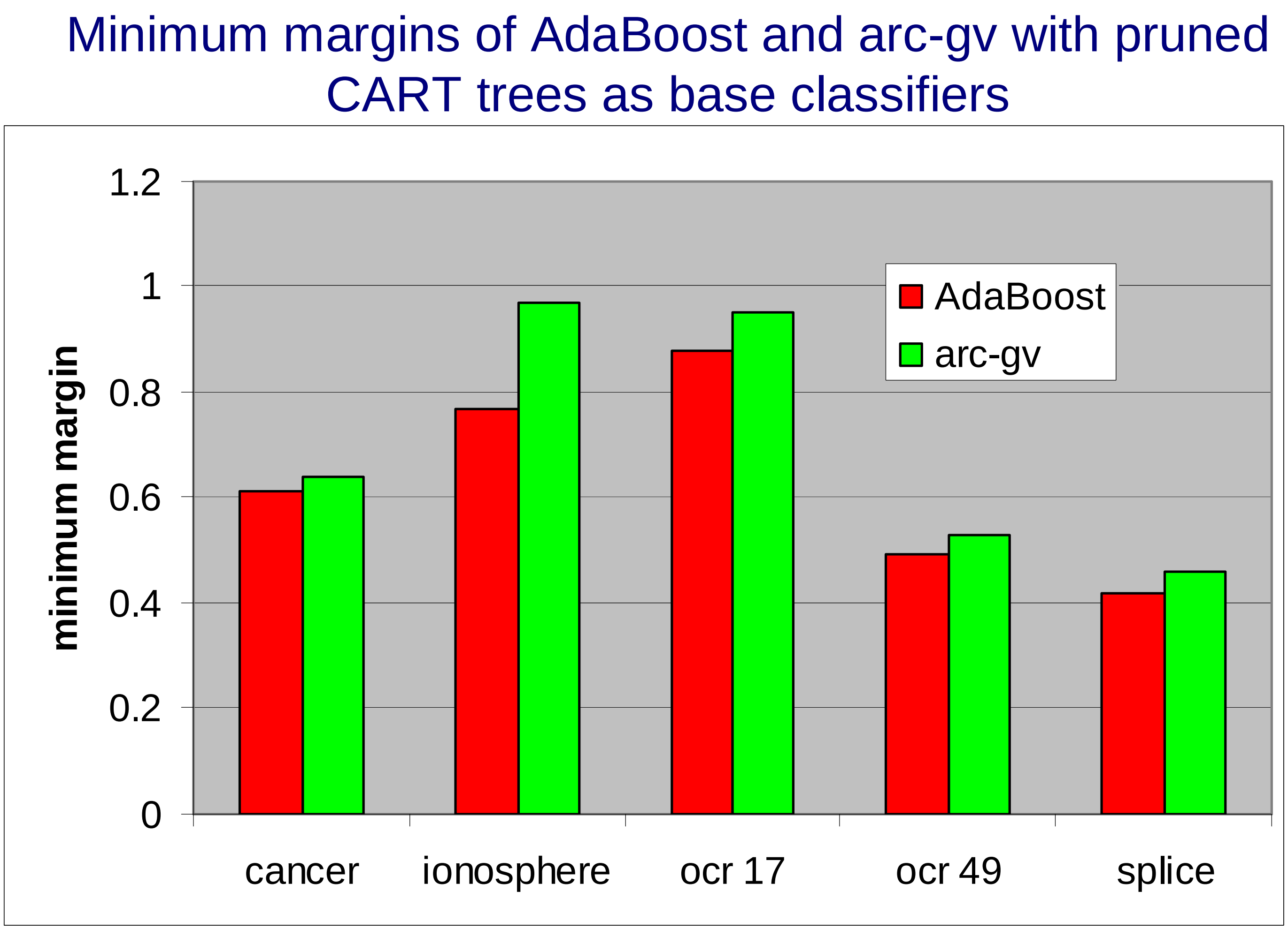
- Data: Breast cancer, ionosphere, and splice
  - From UCI
  - Same natural datasets as Breiman used
- Data: ocr 17, ocr 49
  - Random subsets from NIST
  - Scaled to 14x14 points
- Binary classification
- Use 16-leaf CART trees as base classifiers

Data: the Margins



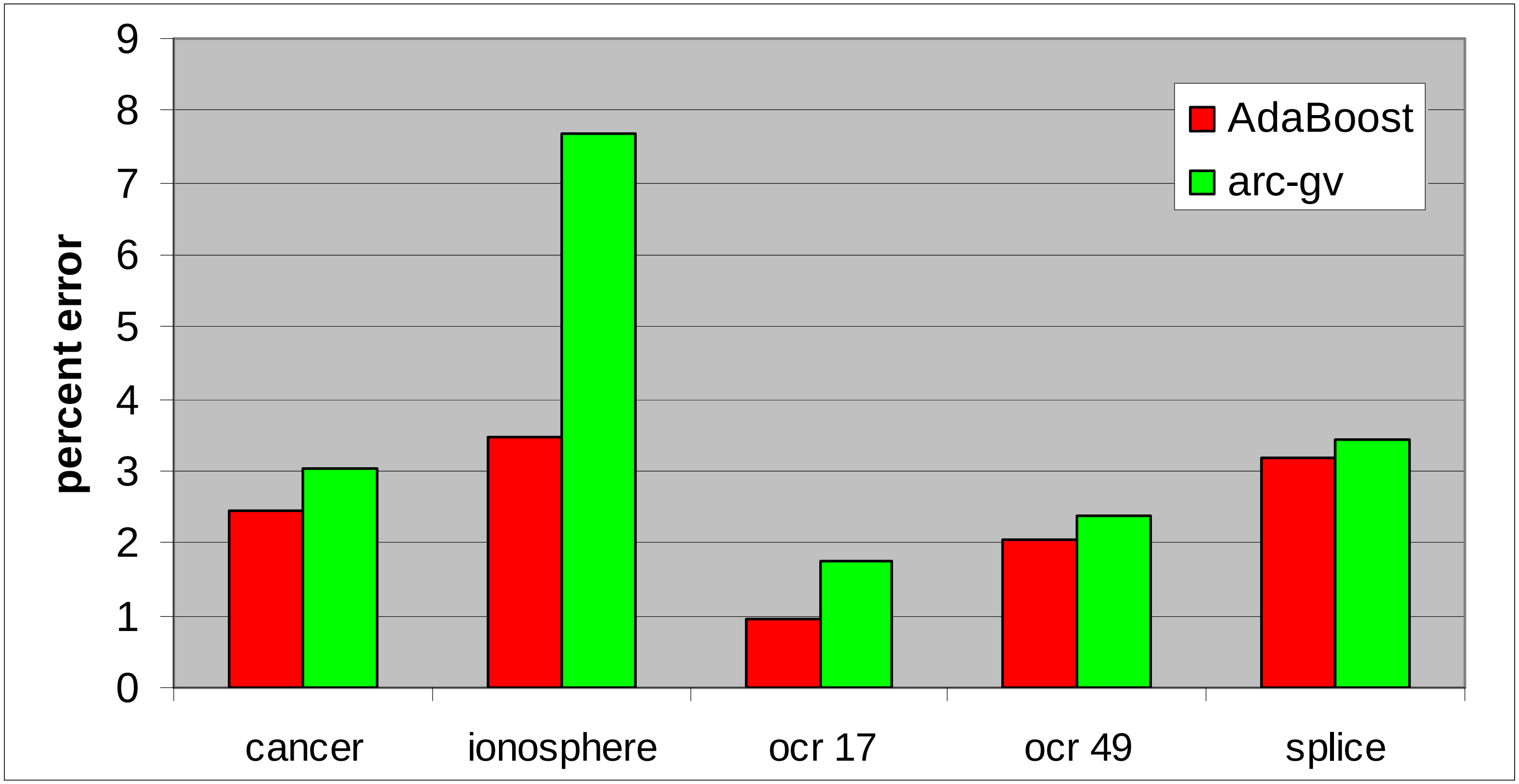
Cumulative margins: 500 rounds of boosting on the “breast cancer” dataset using pruned CART trees as weak learners.

The Minimum Margins



Data: the Errors

Test errors of AdaBoost and arc-gv with pruned CART trees as base classifiers



Doubting the Margins Explanation

- arc-gv has uniformly higher margins than AdaBoost with pruned CART trees.
- the margins explanation predicts that arc-gv should perform better, but instead arc-gv performs worse.
- Breiman's experiment put the margins theory into serious doubt

Reconciling with Margins Theory?

$$\hat{\Pr}\left[\text{margin}_f(x,y) \le \theta\right] + \tilde{O}\left(\sqrt{\frac{d}{m\theta^2}}\right)$$

- Margin bound depends on the entire distribution – not just minimum margin.
  - But arc-gv's margins were uniformly bigger!
- arc-gv may generate bigger, more complex CART trees.
  - But they were pruned to 16 leaves.

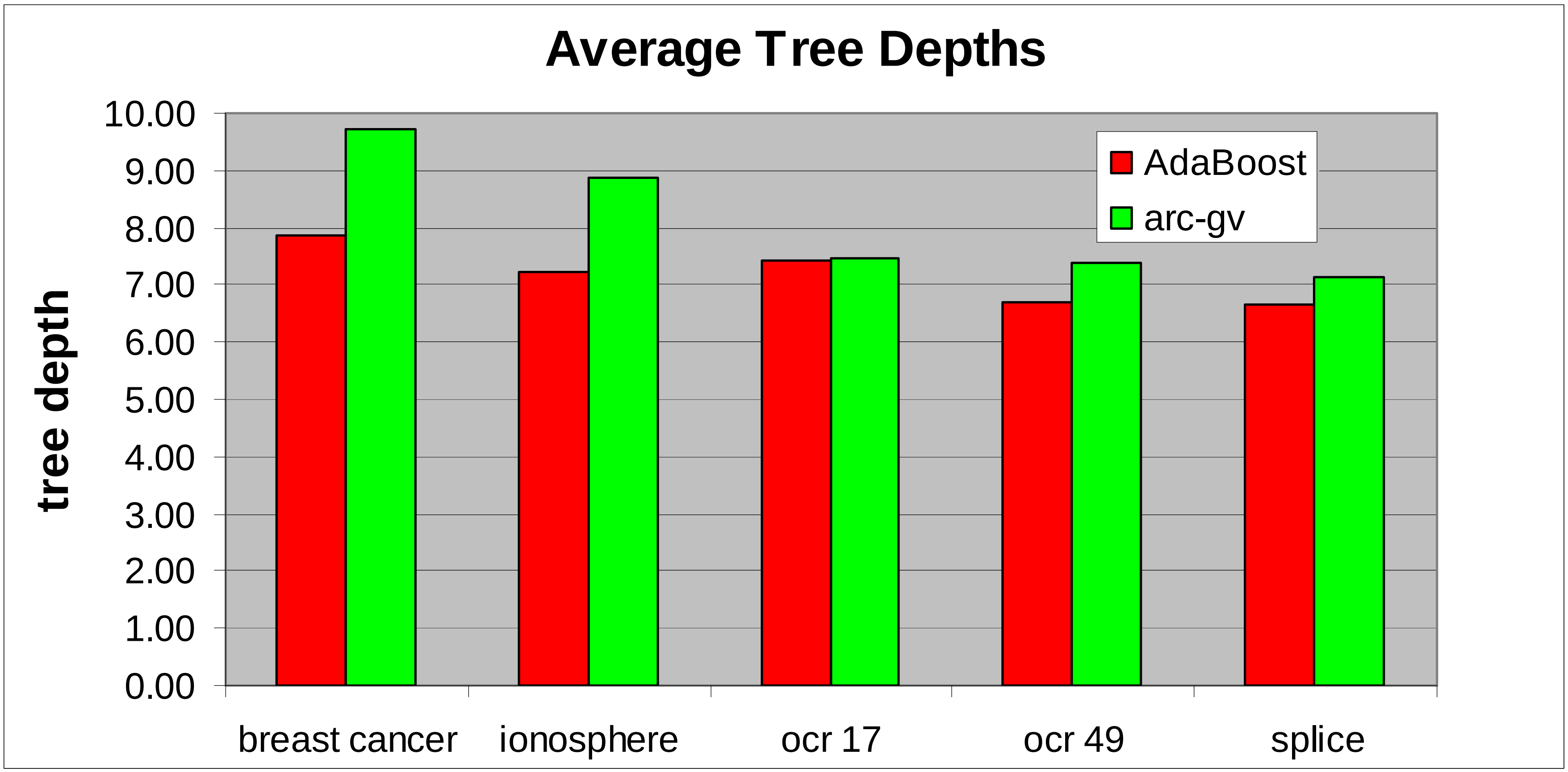
Another Look at the Margins Bound

$$\hat{\Pr}\left[\mathrm{margin}_f(x,y) \le \theta\right] + \tilde{O}\left(\sqrt{\frac{d}{m\theta^2}}\right)$$

- Maybe tree size is too crude a measure of complexity

- Idea: use tree depth as complexity measure
  [Mason et. al. '02]

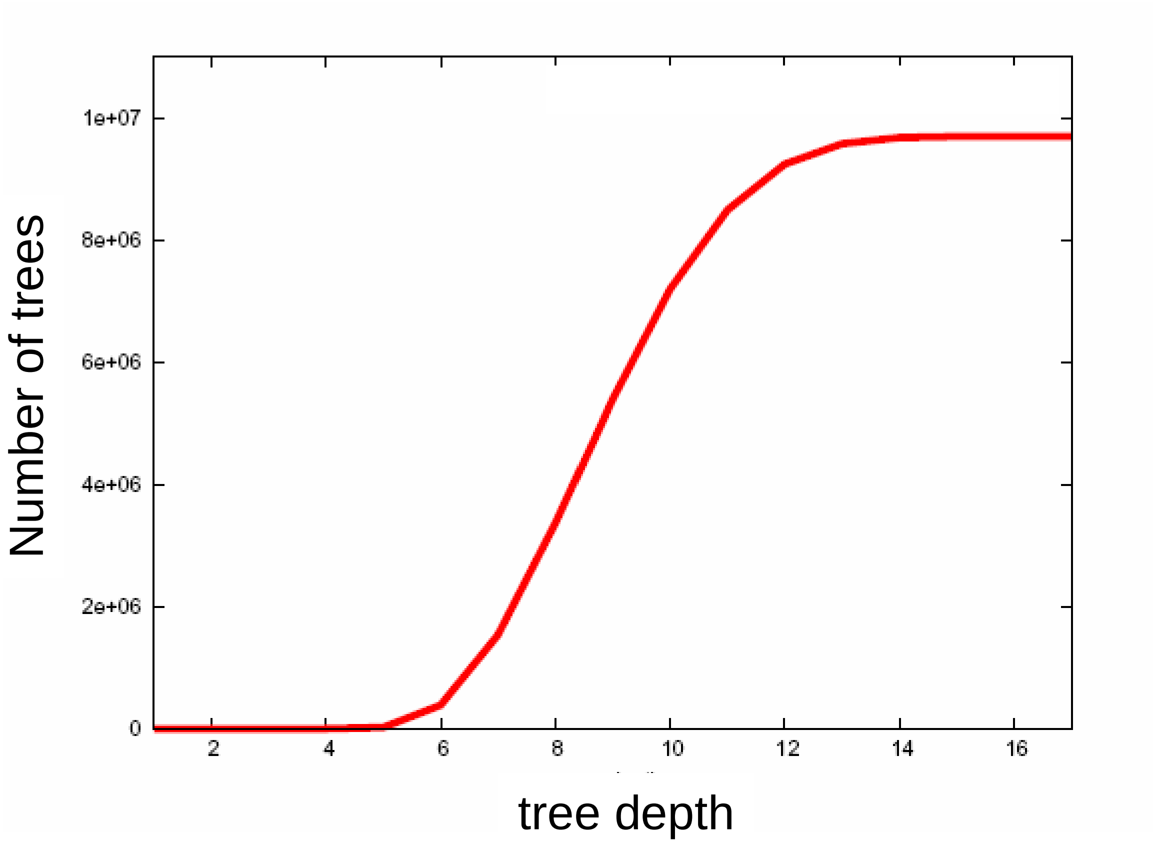
Measuring Tree Depth



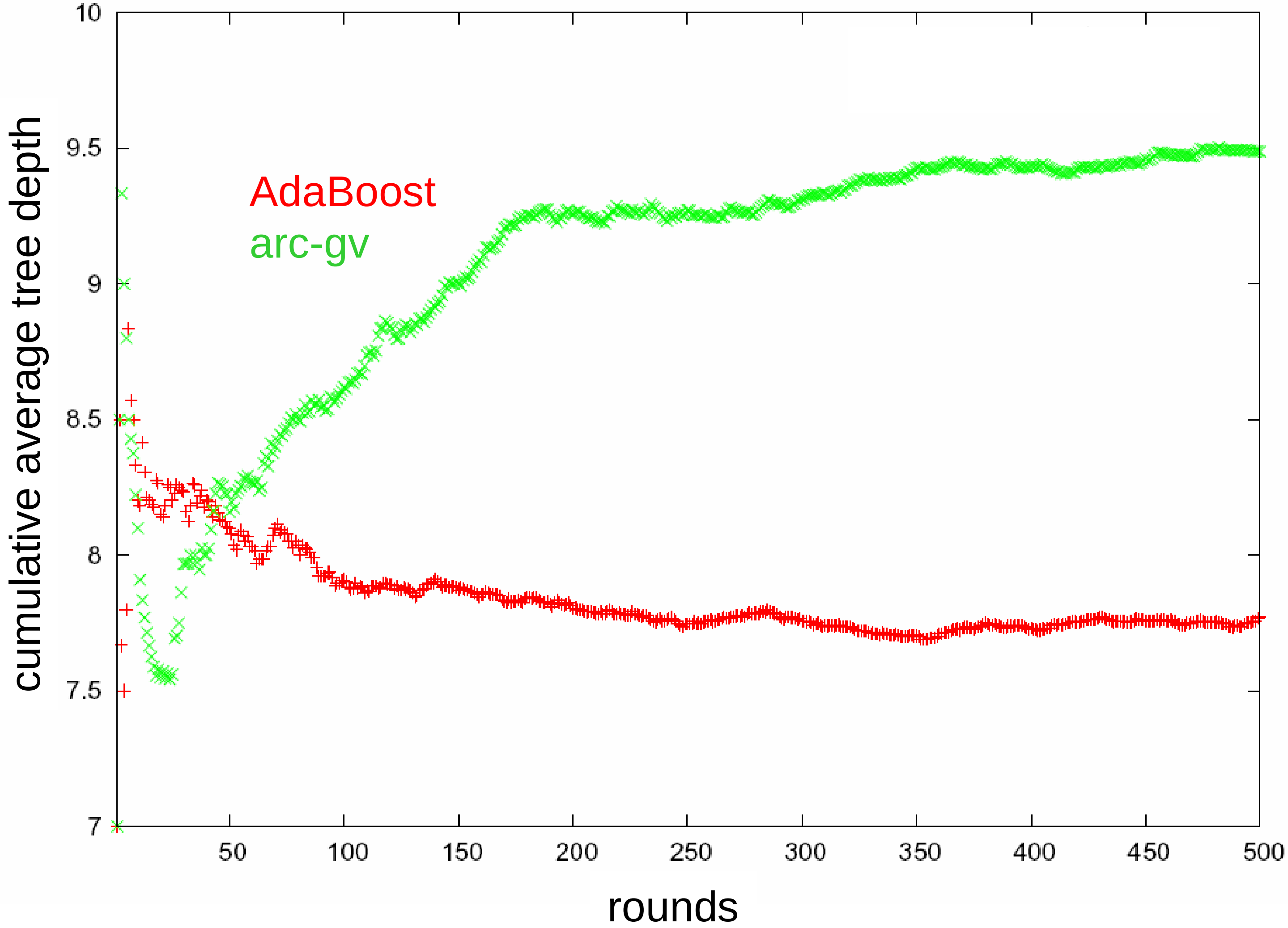
Counting Trees

- We can upper bound the VC-dimension of a finite space of base classifiers H by lg |H|.
- Measuring complexity is essentially a matter of counting how many trees there are of bounded depth.

Trees of Bounded Depth

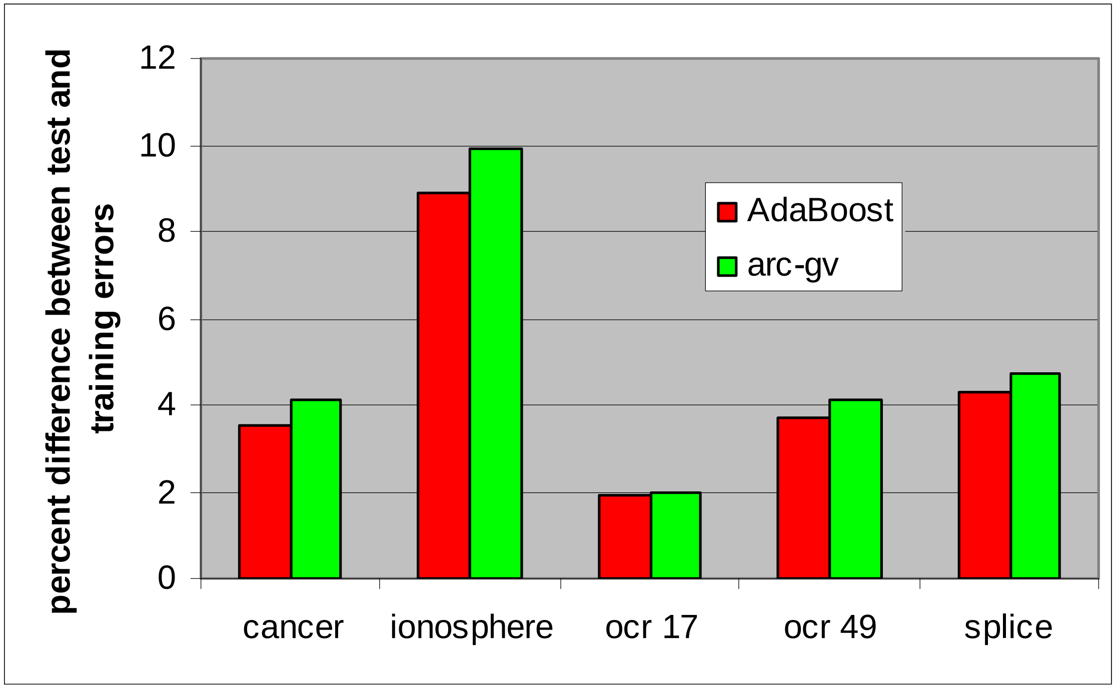


Tree Depth vs Number of Rounds



Another Measure of Tree Complexity

- Idea: difference between training and test error tends to be bigger for higher complexity classifiers.



differences of test and training errors per generated tree averaged over all CART trees in 500 rounds of boosting (over 10 trials)

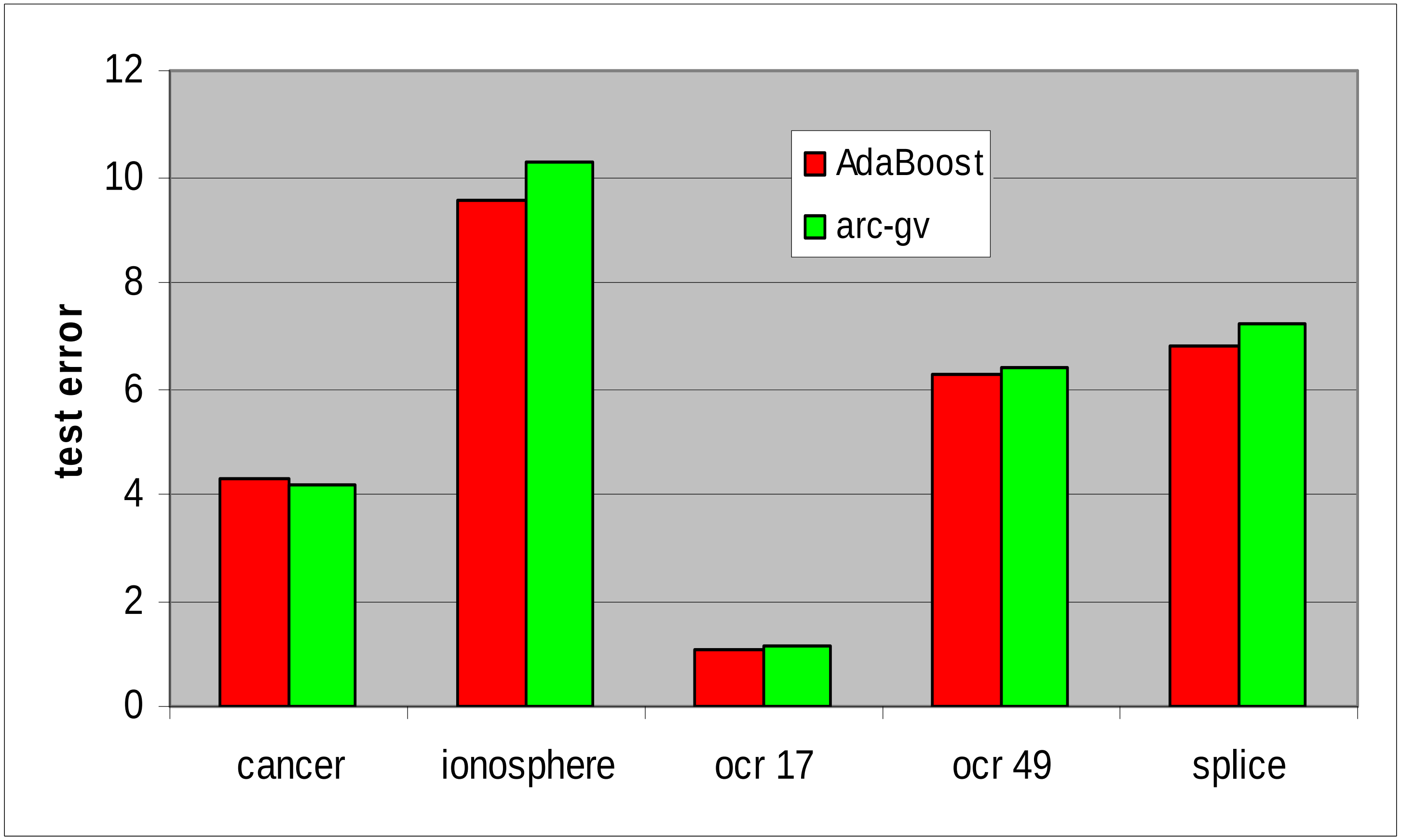
The margins explanation basically says that when all other factors are equal, higher margins result in lower error.

Given that arc-gv tends to choose trees of higher complexity, its higher test error no longer qualitatively contradicts the margin theory.

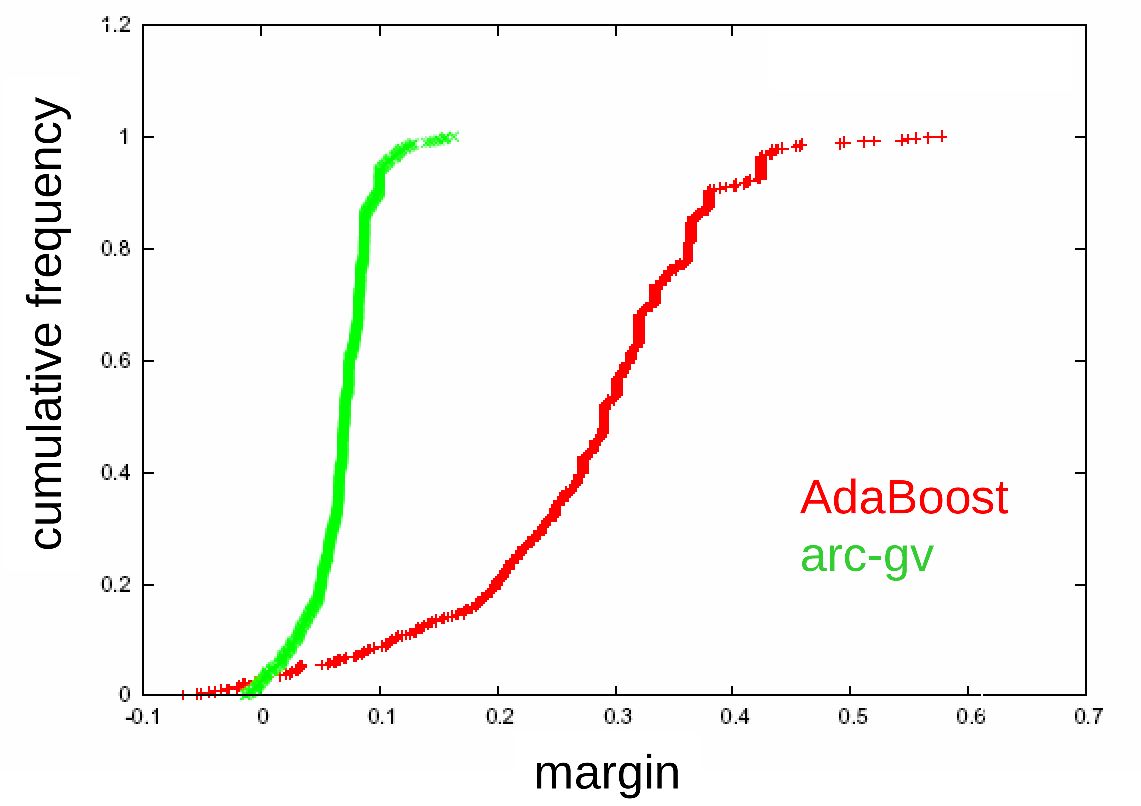
What if we control complexity?

Let's try decision stumps as base classifiers

Controlling Classifier Complexity: Decision Stumps



Decision Stumps



The minimum margin is bigger for arc-gv, but the overall margins distribution is higher for AdaBoost

Discussion

- Breiman's results may not actually contradict the margins theory.
- Margins are important, but not always at the expense of other factors.
- Slightly different boosting algorithms can cause radically different behavior in their generated base classifiers.

Open Questions

- So far, unable to find weak learner of constant complexity with uniformly greater margins distribution for arc-gv than AdaBoost. Does one exist?

- Can we design better boosting algorithms – maximizing average margin?

- Can we prove better (margin) bounds?